

第5章 数组和广义表

DATA STRUCTURE

计算机科学学院 廖雪花

本章内容简介

数组和广义表

5.1 数组的定义

5.2 数组的顺序表示和实现

5.3 矩阵的压缩存储

5.4 广义表的定义

5.5 广义表的存储结构

5.2 数组的顺序表示和实现

廖雪花 LiaoXuehua

数组的顺序表示和实现

■ 存储结构的选择：

- ◆ 由于对数组一般不做插入和删除操作，也就是说，数组一旦建立，结构中的元素个数和元素间的关系就不再发生变化。因此，一般都是采用**顺序存储**的方法来表示数组。
- ◆ 由于**计算机的内存结构是一维的**，因此用一维内存来表示多维数组，就必须按某种次序将数组元素排成一列序列，然后将这个线性序列存放在存储器中。

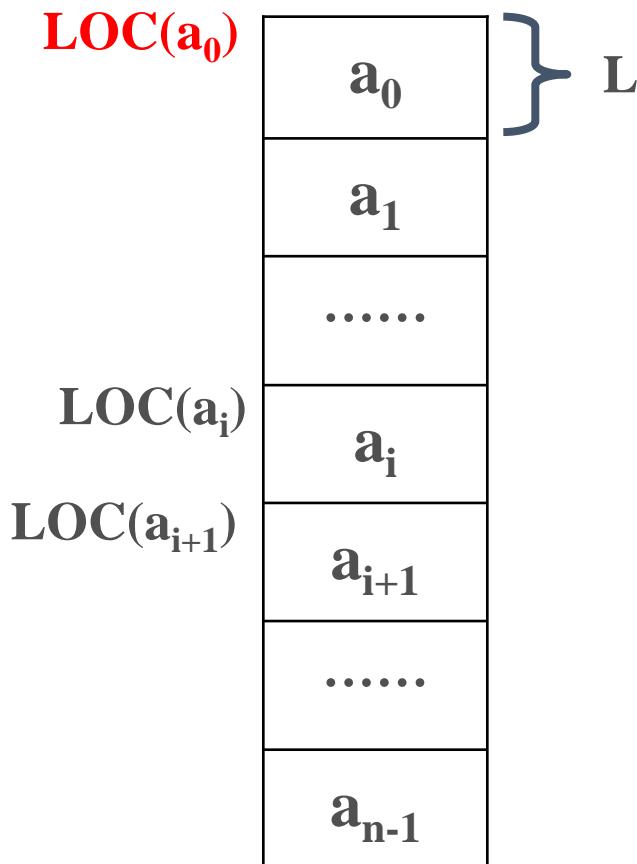
数组的顺序表示和实现

■ 一维数组：

地址计算公式：

$$LOC(a_i) = LOC(a_0) + i \times L$$

$$LOC(a_{i+1}) = LOC(a_i) + L$$



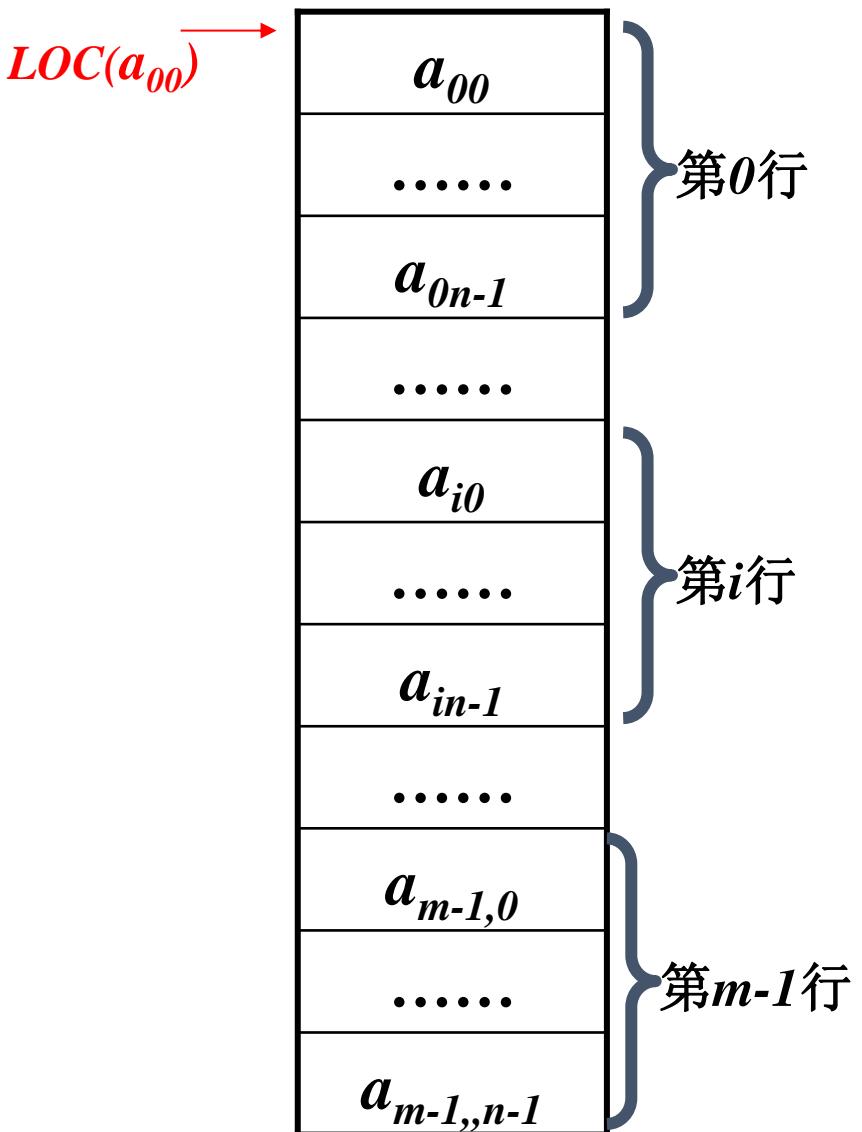
数组的顺序表示和实现

■ 二维数组及多维数组：

(1) 行优先顺序存储

地址计算公式：

$$Loc(a_{ij}) = Loc(a_{00}) + (i \times n + j) L$$



数组的顺序表示和实现

✓例1：

一个二维数组A，行下标的范围是1到6，列下标的范围是0到7，每个数组元素用相邻的6个字节存储，存储器按字节编址。那么，这个数组的体积是(**288**)个字节。

$$6 \times 8 \times 6 = 288$$

数组的顺序表示和实现

✓例2：

设数组a[1...60, 1...70]的基地址为2000，每个元素占2个存储单元，若以行序为主序顺序存储，则元素a[32,31]的存储地址为（**6400**）。

$$2000 + (31 \times 70 + 30) \times 2 = 6400$$

数组的顺序表示和实现

■ 可以推广到多维数组的行优先顺序存储

□ 行优先顺序存储也称为**低下标优先或左边下标优先于右边下标**。

□ 多维数组按行优先存放到内存的规律：

- 最左边下标变化最慢，最右边下标变化最快，右边下标变化一遍，与之相邻的左边下标才变化一次。

$a[\underline{1 \dots 60}, \underline{1 \dots 70}, \underline{1 \dots 80}]$

- 因此，在算法中，最左边下标可以看成是外循环，最右边下标可以看成是最内循环。

数组的顺序表示和实现

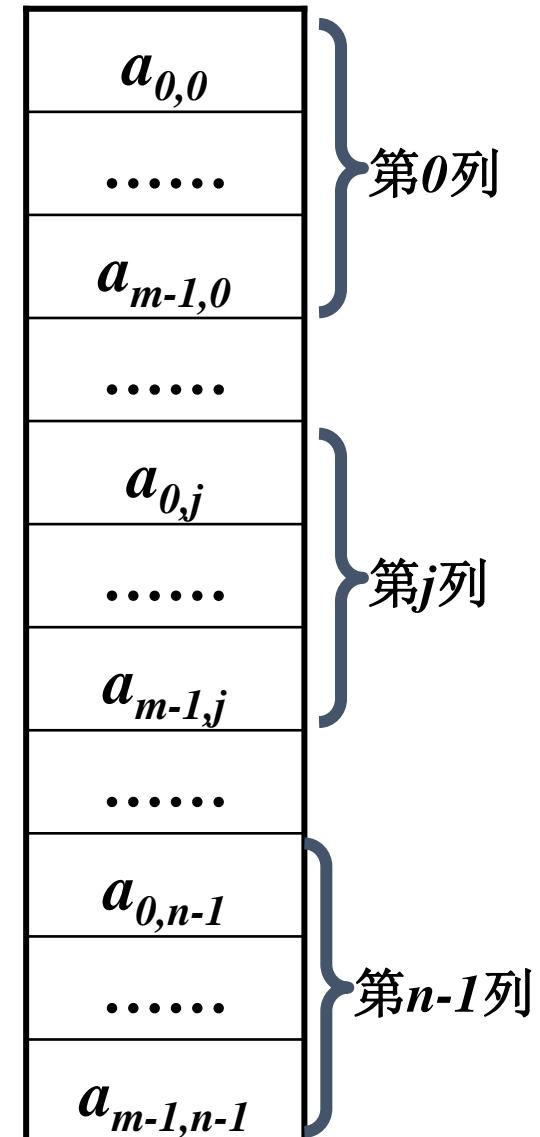
■ 二维数组及多维数组：

(1) 列优先顺序存储

地址计算公式：

$$Loc(a_{ij}) = Loc(a_{00}) + (j \times m + i)L$$

$LOC(a_{00})$



数组的顺序表示和实现

■ 可以推广到多维数组的列优先顺序存储

□ 列优先顺序存储也称为高下标优先或右边下标优先于左边下标。

□ 多维数组按列优先存放到内存的规律：

- 最右边下标变化最慢，最左边下标变化最快，左边下标变化一遍，与之相邻的右边下标才变化一次。

$a[\underline{1 \dots 60}, \underline{1 \dots 70}, \underline{1 \dots 80}]$

- 因此，在算法中，最右边下标可以看成是外循环，最左边下标可以看成是最内循环。

数组的顺序表示和实现

■ 优点：

- ◆ 只要知道以下三要素

- 开始结点的存放地址（即基地址）；
 - 维数和每维的上、下界；
 - 每个数组元素所占用的单元数

就可以将数组元素的存放地址表示为其下标的线性函数。

- ◆ 因此：

- 数组中的任一元素可以在相同的时间内存取，即顺序存储的数组是一个**随机存取**结构。

数组的顺序表示和实现

■ 缺点：

为了在计算机内存中给数组开辟足够的内存空间，必须预先确定每个数组下标的上、下界，有时这是比较困难的。

本节要点

■ 数组的顺序表示和实现：

- ✓ 顺序结构存储
- ✓ 一维数组存储
- ✓ 二维数组的行优先存储和列优先存储
- ✓ 多维数组的行优先存储和列优先存储
- ✓ 数组的顺序存储结构的优缺点
- ✓ 了解数组的逻辑结构和存储表示
- ✓ 掌握数组在以行/列为主的存储结构中的地址计算方法

感谢聆听